

AMENDMENT TO THE CLAIMS

IN THE CLAIMS:

Please amend claims 1-6, 12, 13, 18-23, 29, 30 and 35 as follows. A copy of all pending claims and a status of the claims are provided below.

1. (Currently Amended) A method for creating a data file using a programming development environment on a computer system, said method comprising the steps of:
building a program in said development environment to represent said data file;
compiling the program in said development environment into a software executable; and
running the executable to generate the data file containing definition files which are interpreted wherein the data file is for interpretation by a third party computer system for future application execution, and the third party computer system comprises a dialogue management system for a computer telephony system.
2. (Currently Amended) The method of claim 1 wherein~~whereby~~ the program is built by linking a plurality of development components
3. (Currently Amended) The method of claim 2 wherein~~whereby~~ at least one component comprises characteristic data file information
4. (Currently Amended) The method of claim 3 wherein~~whereby~~, on running the executable, at least one compiled component outputs its respective data file information into the data file.

5. (Currently Amended) The method of claim 4 wherein~~whereby~~, on running the executable, at least one compiled component creates a file output stream and writes its respective data file information to the output stream.
6. (Currently Amended) The method of claim 4 wherein~~whereby~~, on running the executable, at least one compiled component causes another component to output its respective data file information into the data file.
7. (Original) The method of claim 2 wherein at least one development component comprises a graphical icon for a visual development graphical user interface.
8. (Original) The method of claim 2 wherein the development components are Java beans.
9. (Original) The method of claim 2 wherein the development components comprise a main component and a sub-component.
10. (Original) The method of claim 9 wherein the main development component represents a form.
11. (Original) The method of claim 10 wherein the sub-component represents a text field on the form.
12. (Currently Amended) The method of claim 2 wherein~~whereby~~ the program is compiled by generating an executable component from each development component and linking the executable components together.

13. (Currently Amended) The method of claim 12 ~~wherein~~whereby, on running a first executable component, data file information from the first executable is output before running the next and subsequent executable components.

14. (Original) The method of claim 1, wherein the data file comprises mark-up information.

15. (Original) The method of claim 14, wherein the mark-up information comprises XML.

18. (Currently Amended) A system for creating a data file using a programming development environment on a further computer system, said system comprising:
means for building a program in said development environment to represent said data file;

means for compiling the program in said development environment into a software executable; and

means for running the executable to generate the data file containing definition files which are interpreted wherein the data file is for interpretation by a third party computer system for future application execution, and the third party computer system comprises a dialogue management system for a computer telephony system.

19. (Currently Amended) The system of claim 18 ~~wherein~~whereby the program is built by linking a plurality of development components.

20. (Currently Amended) The system of claim 19 ~~wherein~~whereby at least one component comprises characteristic data file information.

21. (Currently Amended) The system of claim 20 wherein~~whereby~~, on running the executable, at least one compiled component outputs its respective data file information into the data file.

22. (Currently Amended) The system of claim 21 wherein~~whereby~~, on running the executable, at least one compiled component creates a file output stream and writes its respective data file information to the output stream.

23. (Currently Amended) The system of claim 21 wherein~~whereby~~, on running the executable, at least one compiled component causes another component to output its respective data file information into the data file.

24. (Original) The system of claim 19 wherein at least one development component comprises a graphical icon for a visual development graphical user interface.

25. (Original) The system of claim 19 wherein the development components are Java beans.

26. (Original) The system of claim 19 wherein the development components comprise a main component and a sub-component.

27. (Original) The system of claim 26 wherein the main development component represents a form.

28. (Original) The system of claim 27 wherein the sub-component represents a text field on the form.

29. (Currently Amended) The system of claim 19 ~~wherein~~whereby the program is compiled by generating an executable component from each development component and linking the executable components together.

30. (Currently Amended) The system of claim 29 ~~wherein~~whereby, on running a first executable component, data file information from the first executable is output before running the next and subsequent executable components.

31. (Original) The system of claim 18, wherein the data file comprises mark-up information.

32. (Original) The system of claim 31, wherein the mark-up information comprises XML.

35. (Currently Amended) A computer program product comprising computer program code stored on a computer readable storage medium for, creating a data file using a programming development environment, said computer program product comprising:
means for building a program in said development environment to represent said data file;

means for compiling the program in said development environment into a software executable; and

means for running the executable to generate the data file containing definition files which are interpreted ~~wherein the data file is for interpretation~~ by a third party computer system for future application execution, and the third party computer system comprises a dialogue management system for a computer telephony system.
